Assignment 4: Data Wrangling (Fall 2024)

Laila Abed

OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics on Data Wrangling

Directions

1. Rename this file `<FirstLast>_A04_DataWrangling.Rmd` (replacing `<FirstLast>` with your first and last name).
2. Change "Student Name" on line 3 (above) with your name.
3. Work through the steps, **creating code and output** that fulfill each instruction.
4. Be sure to **answer the questions** in this assignment document.
5. When you have completed the assignment, **Knit** the text and code into a single PDF file.
6. Ensure that code in code chunks does not extend off the page in the PDF.

Set up your session

1a. Load the `tidyverse`, `lubridate`, and `here` packages into your session.

1b. Check your working directory.

1c. Read in all four raw data files associated with the EPA Air dataset, being sure to set string columns to be read in a factors. See the README file for the EPA air datasets for more information (especially if you have not worked with air quality data previously).

2. Add the appropriate code to reveal the dimensions of the four datasets.

```
#1a
# Upload packages
library(tidyverse)
library(lubridate)
library(here)
#1b
#check work directory
getwd()
```

```
## [1] "/home/guest/EDE_Fall2024"
```

```
here()
```

```
## [1] "/home/guest/EDE_Fall2024"
```

```r
#1c
#Read and save the four data sets.
EPAair_O3_NC2018_raw <- read.csv(
  "/home/guest/EDE_Fall2024/Data/Raw/EPAair_O3_NC2018_raw.csv",
  stringsAsFactors = TRUE
  )
EPAair_O3_NC2019_raw <- read.csv(
  "/home/guest/EDE_Fall2024/Data/Raw/EPAair_O3_NC2019_raw.csv",
  stringsAsFactors = TRUE
  )
EPAair_PM25_NC2018_raw <- read.csv(
  "/home/guest/EDE_Fall2024/Data/Raw/EPAair_PM25_NC2018_raw.csv",
  stringsAsFactors = TRUE
  )
EPAair_PM25_NC2019_raw <- read.csv(
  "/home/guest/EDE_Fall2024/Data/Raw/EPAair_PM25_NC2019_raw.csv",
  stringsAsFactors = TRUE
  )
  # have a peak to the data sets
glimpse(EPAair_O3_NC2018_raw)
```

```
## Rows: 9,737
## Columns: 20
## $ Date                                 <fct> 03/01/2018, 03/02/2018, 03/03/201~
## $ Source                               <fct> AQS, AQS, AQS, AQS, AQS, AQS, AQS~
## $ Site.ID                              <int> 370030005, 370030005, 370030005, ~
## $ POC                                  <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ Daily.Max.8.hour.Ozone.Concentration <dbl> 0.043, 0.046, 0.047, 0.049, 0.047~
## $ UNITS                                <fct> ppm, ppm, ppm, ppm, ppm, ppm, ppm~
## $ DAILY_AQI_VALUE                      <int> 40, 43, 44, 45, 44, 28, 33, 41, 4~
## $ Site.Name                            <fct> Taylorsville Liledoun, Taylorsvil~
## $ DAILY_OBS_COUNT                      <int> 17, 17, 17, 17, 17, 17, 17, 17, 1~
## $ PERCENT_COMPLETE                     <dbl> 100, 100, 100, 100, 100, 100, 100~
## $ AQS_PARAMETER_CODE                   <int> 44201, 44201, 44201, 44201, 44201~
## $ AQS_PARAMETER_DESC                   <fct> Ozone, Ozone, Ozone, Ozone, Ozone~
## $ CBSA_CODE                            <int> 25860, 25860, 25860, 25860, 25860~
## $ CBSA_NAME                            <fct> "Hickory-Lenoir-Morganton, NC", "~
## $ STATE_CODE                           <int> 37, 37, 37, 37, 37, 37, 37, 37, 3~
## $ STATE                                <fct> North Carolina, North Carolina, N~
## $ COUNTY_CODE                          <int> 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, ~
## $ COUNTY                               <fct> Alexander, Alexander, Alexander, ~
## $ SITE_LATITUDE                        <dbl> 35.9138, 35.9138, 35.9138, 35.913~
## $ SITE_LONGITUDE                       <dbl> -81.191, -81.191, -81.191, -81.19~
```

```r

```

```
## Rows: 10,592
## Columns: 20
## $ Date                                 <fct> 01/01/2019, 01/02/2019, 01/03/201~
## $ Source                               <fct> AirNow, AirNow, AirNow, AirNow, A~
## $ Site.ID                              <int> 370030005, 370030005, 370030005, ~
## $ POC                                  <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ Daily.Max.8.hour.Ozone.Concentration <dbl> 0.029, 0.018, 0.016, 0.022, 0.037~
```

```
## $ UNITS                          <fct> ppm, ppm, ppm, ppm, ppm, ppm, ppm~
## $ DAILY_AQI_VALUE                <int> 27, 17, 15, 20, 34, 34, 27, 35, 3~
## $ Site.Name                      <fct> Taylorsville Liledoun, Taylorsvil~
## $ DAILY_OBS_COUNT                <int> 24, 24, 24, 24, 24, 24, 24, 24, 2~
## $ PERCENT_COMPLETE               <dbl> 100, 100, 100, 100, 100, 100, 100~
## $ AQS_PARAMETER_CODE             <int> 44201, 44201, 44201, 44201, 44201~
## $ AQS_PARAMETER_DESC             <fct> Ozone, Ozone, Ozone, Ozone, Ozone~
## $ CBSA_CODE                      <int> 25860, 25860, 25860, 25860, 25860~
## $ CBSA_NAME                      <fct> "Hickory-Lenoir-Morganton, NC", "~
## $ STATE_CODE                     <int> 37, 37, 37, 37, 37, 37, 37, 37, 3~
## $ STATE                          <fct> North Carolina, North Carolina, N~
## $ COUNTY_CODE                    <int> 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, ~
## $ COUNTY                         <fct> Alexander, Alexander, Alexander, ~
## $ SITE_LATITUDE                  <dbl> 35.9138, 35.9138, 35.9138, 35.913~
## $ SITE_LONGITUDE                 <dbl> -81.191, -81.191, -81.191, -81.19~
```

```
## Rows: 8,983
## Columns: 20
## $ Date                           <fct> 01/02/2018, 01/05/2018, 01/08/2018, 01/~
## $ Source                         <fct> AQS, AQS, AQS, AQS, AQS, AQS, AQS, AQS,~
## $ Site.ID                        <int> 370110002, 370110002, 370110002, 370110~
## $ POC                            <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ Daily.Mean.PM2.5.Concentration <dbl> 2.9, 3.7, 5.3, 0.8, 2.5, 4.5, 1.8, 2.5,~
## $ UNITS                          <fct> ug/m3 LC, ug/m3 LC, ug/m3 LC, ug/m3 LC,~
## $ DAILY_AQI_VALUE                <int> 12, 15, 22, 3, 10, 19, 8, 10, 18, 7, 24~
## $ Site.Name                      <fct> Linville Falls, Linville Falls, Linvill~
## $ DAILY_OBS_COUNT                <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ PERCENT_COMPLETE               <dbl> 100, 100, 100, 100, 100, 100, 100, 100,~
## $ AQS_PARAMETER_CODE             <int> 88502, 88502, 88502, 88502, 88502, 8850~
## $ AQS_PARAMETER_DESC             <fct> Acceptable PM2.5 AQI & Speciation Mass,~
## $ CBSA_CODE                      <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ CBSA_NAME                      <fct> "", "", "", "", "", "", "", "", "", "",~
## $ STATE_CODE                     <int> 37, 37, 37, 37, 37, 37, 37, 37, 37, 37,~
## $ STATE                          <fct> North Carolina, North Carolina, North C~
## $ COUNTY_CODE                    <int> 11, 11, 11, 11, 11, 11, 11, 11, 11, 11,~
## $ COUNTY                         <fct> Avery, Avery, Avery, Avery, Avery, Aver~
## $ SITE_LATITUDE                  <dbl> 35.97235, 35.97235, 35.97235, 35.97235,~
## $ SITE_LONGITUDE                 <dbl> -81.93307, -81.93307, -81.93307, -81.93~
```

```
## Rows: 8,581
## Columns: 20
## $ Date                           <fct> 01/03/2019, 01/06/2019, 01/09/2019, 01/~
## $ Source                         <fct> AQS, AQS, AQS, AQS, AQS, AQS, AQS, AQS,~
## $ Site.ID                        <int> 370110002, 370110002, 370110002, 370110~
## $ POC                            <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ Daily.Mean.PM2.5.Concentration <dbl> 1.6, 1.0, 1.3, 6.3, 2.6, 1.2, 1.5, 1.5,~
## $ UNITS                          <fct> ug/m3 LC, ug/m3 LC, ug/m3 LC, ug/m3 LC,~
## $ DAILY_AQI_VALUE                <int> 7, 4, 5, 26, 11, 5, 6, 6, 15, 7, 14, 20~
## $ Site.Name                      <fct> Linville Falls, Linville Falls, Linvill~
```

```
## $ DAILY_OBS_COUNT                <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ PERCENT_COMPLETE               <dbl> 100, 100, 100, 100, 100, 100, 100, 100,~
## $ AQS_PARAMETER_CODE             <int> 88502, 88502, 88502, 88502, 88502, 8850~
## $ AQS_PARAMETER_DESC             <fct> Acceptable PM2.5 AQI & Speciation Mass,~
## $ CBSA_CODE                      <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ CBSA_NAME                      <fct> "", "", "", "", "", "", "", "", "", "",~
## $ STATE_CODE                     <int> 37, 37, 37, 37, 37, 37, 37, 37, 37, 37,~
## $ STATE                          <fct> North Carolina, North Carolina, North C~
## $ COUNTY_CODE                    <int> 11, 11, 11, 11, 11, 11, 11, 11, 11, 11,~
## $ COUNTY                         <fct> Avery, Avery, Avery, Avery, Avery, Aver~
## $ SITE_LATITUDE                  <dbl> 35.97235, 35.97235, 35.97235, 35.97235,~
## $ SITE_LONGITUDE                 <dbl> -81.93307, -81.93307, -81.93307, -81.93~
```

```r
#2
# check the dimensions of the four data sets
dim(EPAair_O3_NC2018_raw)
```

```
## [1] 9737   20
```

```r
dim(EPAair_O3_NC2019_raw)
```

```
## [1] 10592    20
```

```r
dim(EPAair_PM25_NC2018_raw)
```

```
## [1] 8983   20
```

```r
dim(EPAair_PM25_NC2019_raw)
```

```
## [1] 8581   20
```

> All four datasets should have the same number of columns but unique record counts (rows). Do your datasets follow this pattern? Yes, correct. All the four have 20 columns but different rows (9737, 10592, 8983, 8581)

Wrangle individual datasets to create processed files.

3. Change the Date columns to be date objects.
4. Select the following columns: Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC, COUNTY, SITE_LATITUDE, SITE_LONGITUDE
5. For the PM2.5 datasets, fill all cells in AQS_PARAMETER_DESC with "PM2.5" (all cells in this column should be identical).
6. Save all four processed datasets in the Processed folder. Use the same file names as the raw files but replace "raw" with "processed".

```r
#3
# Change the Date columns to be date objects
EPAair_O3_NC2018_raw$Date <- as.Date(
  EPAair_O3_NC2018_raw$Date, format = "%m/%d/%Y")
EPAair_O3_NC2019_raw$Date <- as.Date(
  EPAair_O3_NC2019_raw$Date, format = "%m/%d/%Y")
EPAair_PM25_NC2018_raw$Date <- as.Date(
  EPAair_PM25_NC2018_raw$Date, format = "%m/%d/%Y")
EPAair_PM25_NC2019_raw$Date <- as.Date(
  EPAair_PM25_NC2019_raw$Date, format = "%m/%d/%Y")

#4
# make new data sets with the following columns: Date, DAILY_AQI_VALUE,
#Site.Name, AQS_PARAMETER_DESC, COUNTY, SITE_LATITUDE, SITE_LONGITUDE.
EPAair_O3_NC2018_Subdata <- select(
  EPAair_O3_NC2018_raw, Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC,
  COUNTY, SITE_LATITUDE, SITE_LONGITUDE)
EPAair_O3_NC2019_Subdata <- select(EPAair_O3_NC2019_raw, Date, DAILY_AQI_VALUE,
Site.Name, AQS_PARAMETER_DESC, COUNTY, SITE_LATITUDE, SITE_LONGITUDE)
EPAair_PM25_NC2018_Subdata <- select(EPAair_PM25_NC2018_raw, Date,
DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC, COUNTY, SITE_LATITUDE,
SITE_LONGITUDE)
EPAair_PM25_NC2019_Subdata <- select(EPAair_PM25_NC2019_raw, Date,
DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC, COUNTY, SITE_LATITUDE,
SITE_LONGITUDE)
#5
# For the PM2.5 2018 and 2019, fill all cells in AQS_PARAMETER_DESC with "PM2.5"
EPAair_PM25_NC2018_Subdata <- EPAair_PM25_NC2018_Subdata %>%
mutate(AQS_PARAMETER_DESC = "PM2.5")
EPAair_PM25_NC2019_Subdata <- EPAair_PM25_NC2019_Subdata %>%
  mutate(AQS_PARAMETER_DESC = "PM2.5")

#6
#Save all four processed datasets in the Processed folder
write.csv(EPAair_O3_NC2018_Subdata, row.names = FALSE, file =  "/home/guest/EDE_Fall2024/Data/Processed/
          )
write.csv(EPAair_O3_NC2019_Subdata, row.names = FALSE, file =  "/home/guest/EDE_Fall2024/Data/Processed/
          )
write.csv(EPAair_PM25_NC2018_Subdata, row.names = FALSE, file =  "/home/guest/EDE_Fall2024/Data/Processe
          )
write.csv(EPAair_PM25_NC2019_Subdata, row.names = FALSE, file =  "/home/guest/EDE_Fall2024/Data/Processe
          )
```

Combine datasets

7. Combine the four datasets with `rbind`. Make sure your column names are identical prior to running this code.

8. Wrangle your new dataset with a pipe function (%>%) so that it fills the following conditions:

- Include only sites that the four data frames have in common:

“Linville Falls”, “Durham Armory”, “Leggett”, “Hattie Avenue”, "Clemmons Middle", "Mendenhall School", "Frying Pan Mountain", "West Johnston Co.", "Garinger High School", "Castle Hayne", "Pitt Agri. Center", "Bryson City", "Millbrook School"

(the function `intersect` can figure out common factor levels - but it will include sites with missing site information, which you don't want...)

- Some sites have multiple measurements per day. Use the split-apply-combine strategy to generate daily means: group by date, site name, AQS parameter, and county. Take the mean of the AQI value, latitude, and longitude.
- Add columns for "Month" and "Year" by parsing your "Date" column (hint: `lubridate` package)
- Hint: the dimensions of this dataset should be 14,752 x 9.

9. Spread your datasets such that AQI values for ozone and PM2.5 are in separate columns. Each location on a specific date should now occupy only one row.
10. Call up the dimensions of your new tidy dataset.
11. Save your processed dataset with the following file name: "EPAair_O3_PM25_NC1819_Processed.csv"

```r
#7
#Check the columns of the four datasets to make sure they are in the same order.
colnames(EPAair_O3_NC2018_Subdata)
```

```
## [1] "Date"               "DAILY_AQI_VALUE"    "Site.Name"
## [4] "AQS_PARAMETER_DESC" "COUNTY"             "SITE_LATITUDE"
## [7] "SITE_LONGITUDE"
```

```r
colnames(EPAair_O3_NC2019_Subdata)
```

```
## [1] "Date"               "DAILY_AQI_VALUE"    "Site.Name"
## [4] "AQS_PARAMETER_DESC" "COUNTY"             "SITE_LATITUDE"
## [7] "SITE_LONGITUDE"
```

```r
colnames(EPAair_PM25_NC2018_Subdata)
```

```
## [1] "Date"               "DAILY_AQI_VALUE"    "Site.Name"
## [4] "AQS_PARAMETER_DESC" "COUNTY"             "SITE_LATITUDE"
## [7] "SITE_LONGITUDE"
```

```r
colnames(EPAair_PM25_NC2019_Subdata)
```

```
## [1] "Date"               "DAILY_AQI_VALUE"    "Site.Name"
## [4] "AQS_PARAMETER_DESC" "COUNTY"             "SITE_LATITUDE"
## [7] "SITE_LONGITUDE"
```

```r
#Combine the four datasets
combined_dataset <- rbind(
  EPAair_O3_NC2018_Subdata, EPAair_O3_NC2019_Subdata,
  EPAair_PM25_NC2018_Subdata, EPAair_PM25_NC2019_Subdata)
```

```r
#8
# Upload dplyr
library(dplyr)
# Make a vector with the the common site names.
common_sites <- c(
"Linville Falls", "Durham Armory", "Leggett", "Hattie Avenue",
"Clemmons Middle", "Mendenhall School", "Frying Pan Mountain",
"West Johnston Co.", "Garinger High School", "Castle Hayne",
"Pitt Agri. Center", "Bryson City", "Millbrook School"
)
# Filter the combined data set for the common site names identified.
filtered_dataset <- combined_dataset %>%
filter(Site.Name %in% common_sites)

grouped_dataset <- filtered_dataset %>%
group_by(Date, Site.Name, AQS_PARAMETER_DESC, COUNTY) %>%
summarize(
meanAQI = mean(DAILY_AQI_VALUE),
meanLAT = mean(SITE_LATITUDE),
meanLONG = mean(SITE_LONGITUDE),
)
```

```
## `summarise()` has grouped output by 'Date', 'Site.Name', 'AQS_PARAMETER_DESC'.
## You can override using the `.groups` argument.
```

```r
grouped_dataset <- grouped_dataset %>%
mutate(
Month = month(Date),
Year = year(Date)
)
#check the dimensions
dim(grouped_dataset)
```

```
## [1] 14752     9
```

```r
#drop.na(dataset_common_sites)
#9
# Spread datasets so AQI values for ozone and PM2.5 are in separate columns.
spread_dataset <- grouped_dataset %>%
pivot_wider(
id_cols = c(Date, Site.Name, COUNTY, Year, Month, meanLAT, meanLONG),
names_from = AQS_PARAMETER_DESC,
values_from = meanAQI,
names_prefix = "AQI_"
)

# have a look at the dataset.
glimpse(spread_dataset)
```

```
## Rows: 8,976
## Columns: 9
## Groups: Date, Site.Name [8,976]
```

```
## $ Date      <date> 2018-01-01, 2018-01-01, 2018-01-01, 2018-01-01, 2018-01-01,~
## $ Site.Name <fct> Bryson City, Castle Hayne, Clemmons Middle, Durham Armory, G~
## $ COUNTY    <fct> Swain, New Hanover, Forsyth, Durham, Mecklenburg, Forsyth, E~
## $ Year      <dbl> 2018, 2018, 2018, 2018, 2018, 2018, 2018, 2018, 2018, 2018, ~
## $ Month     <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ meanLAT   <dbl> 35.43477, 34.36417, 36.02600, 36.03296, 35.24010, 36.11069, ~
## $ meanLONG  <dbl> -83.44213, -77.83861, -80.34200, -78.90404, -80.78568, -80.2~
## $ AQI_PM2.5 <dbl> 35.0, 13.0, 24.0, 31.0, 20.0, 22.0, 14.0, 28.0, 15.0, 24.0, ~
## $ AQI_Ozone <dbl> NA, NA, NA, NA, 32, NA, NA, 34, NA, NA, NA, NA, NA, NA, NA, ~
```

```r
#10
# check data set dimensions.
dim(spread_dataset)
```

```
## [1] 8976    9
```

```r
#11
# Save the processed data set.
write.csv(spread_dataset, file.path(
  "/home/guest/EDE_Fall2024/Data/Processed/EPAair_O3_PM25_NC1819_Processed.csv"
  ), row.names = FALSE)
```

Generate summary tables

12. Use the split-apply-combine strategy to generate a summary data frame. Data should be grouped by site, month, and year. Generate the mean AQI values for ozone and PM2.5 for each group. Then, add a pipe to remove instances where mean **ozone** values are not available (use the function `drop_na` in your pipe). It's ok to have missing mean PM2.5 values in this result.
13. Call up the dimensions of the summary dataset.

```r
#12
# generate a summary data frame.
summary_dataset <- spread_dataset %>%
group_by(Site.Name, Year, Month) %>%

summarize(
mean_AQI_Ozone = mean(AQI_Ozone, na.rm = TRUE),
mean_AQI_PM25 = mean(AQI_PM2.5, na.rm = TRUE)
) %>%
# Then I remove instances where mean ozone values are not available
drop_na(mean_AQI_Ozone)
```

```
## `summarise()` has grouped output by 'Site.Name', 'Year'. You can override using
## the `.groups` argument.
```

```r
#na.omit(mean_AQI_Ozone)
# have a look at the data set
head(summary_dataset)
```

```
## # A tibble: 6 x 5
## # Groups:   Site.Name, Year [1]
##   Site.Name    Year Month mean_AQI_Ozone mean_AQI_PM25
##   <fct>       <dbl> <dbl>          <dbl>         <dbl>
## 1 Bryson City  2018     3           41.6          34.7
## 2 Bryson City  2018     4           44.5          28.2
## 3 Bryson City  2018     5           35.9          33.5
## 4 Bryson City  2018     6           37.8          25.1
## 5 Bryson City  2018     7           34.6          34.3
## 6 Bryson City  2018     8           30.8          32.7
```

```r
#13
#Chack the dimensions of teh data set.
dim(summary_dataset)
```

```
## [1] 239   5
```

14. Why did we use the function `drop_na` rather than `na.omit`? Hint: replace `drop_na` with `na.omit` in part 12 and observe what happens with the dimensions of the summary date frame.

    Answer: We used the drop_na function rather than na.omit because drop_na allows us to drop the NA rows in specific columns (gives us flexiability and customization), while na.omit remove all NA in the entire data frame changing the total number of rows. In this problem set, we want to keep rows with missing PM2.5 values but removing the rows with missing ozone values that's why we used drop na.